UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/548,736	04/13/2000	Qinghua Zheng	10001205-1	6965

22879 7590 12/24/2003

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER	
GARCIA OTERO, EDUARDO	
ART UNIT	**PAPER NUMBER**
2123	2

DATE MAILED: 12/24/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/548,736	ZHENG, QINGHUA	
	Examiner	Art Unit	
	Eduardo Garcia-Otero	2123	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *13 April 2000*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. §§ 119 and 120

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

13)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

a) ☐ The translation of the foreign language provisional application has been received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION: Non-Final (first action on the merits)

Introduction

1. Title is: Method and Apparatus for Validating Cross-Architecture ISA Emulation.
2. First named inventor is: ZENG.
3. Claims 1-20 have been submitted, examined, and rejected.
4. US Application filed on 4/13/00, and no earlier priority is claimed.

Index of Prior Art

5. **Mitchell** refers to US Patent 4,841,476.
6. **Banks** refers to Handbook of Simulation, by Jerry Banks, John Wiley & Sons, Inc., August 1998, ISBN 0-471-13403-1, pages 3, 15-19.
7. **Aharon** refers to US Patent 5,202,889.
8. **Tucker** refers to The Computer Science and Engineering Handbook, by Allen B. Tucker, CRC Press, ISBN: 0-8493-2909-4, 1996, pages 412-415.
9. **Gowin** refers to US Patent 6,606,721 which claims priority to provisional application 60/165,204 filed Nov. 12, 1999.
10. **Scalzi** refers to US Patent 6,009,261.

Definitions

11. **Freedman** refers to The Computer Desktop Encyclopedia by Alan Freedman, The Computer Language Company Inc., 1996. ISBN 0-8144-0010-8.

    EMULATOR: "A device that is built to work like another. A computer can be designed to emulate another model and execute software that was written to run in the other machine. A terminal can be designed to emulate various communications protocols and connect to different networks. The emulator can be hardware, software or both."

    ERROR HANDLING: "Routines in a program that respond to errors. The measurement of quality in error handling is based on how the system informs the user of such conditions and what alternatives it provides for dealing with them".

    INSTRUCTION SET: "The repertoire of machine language instructions that a computer can follow (from a handful to several hundred). It is a major architectural component and is either built into the CPU or into microcode. Instructions are generally from one to four bytes long."

ISA: "(1) (Industry Standard Architecture)... An expansion bus commonly used in PCs.... (2) (Interactive Services Association) A trade group for the online industry...".

SIMULATION: "(1) The mathematical representation of the interaction of real-world objects. *See scientific applications*. (2) The execution of a machine language program designed to run in a foreign computer." Italics in original.

TCP/IP: "(Transmission Control Protocol/Internet Protocol) A communications protocol... to internetwork dissimilar systems. It is a de facto UNIX standard, but is supported on almost all platforms. TCP/IP is the protocol of the Internet".

12. **McGraw-Hill Dictionary** refers to The McGraw-Hill Dictionary of Scientific and Technical Terms, Fourth Edition, by McGraw-Hill Companies, Inc., ISBN 0-07-05270-9, 1989.

    EMULATION: "[COMPUT SCI] Imitation of one computer system by another so that the latter functions in exactly the same way and runs the same programs."

    EMULATION MODE: "[COMPUT SCI] A method of operation in which a computer actually executes the instructions of a different (simpler) computer, in contrast to normal model."

    EMULATOR: "[COMPUT SCI] The microprogram-assisted macroprogram which allows a computer to run programs written for another computer."

    EMULATOR CIRCUIT: [COMPUT SCI] A circuit built into a computer's control section to enable it to process instructions that were written for another computer."

    SIMULATE: "[ENG] To mimic some or all of the behavior of one system with a different dissimilar system, particularly with computers, models, or equipment."

13. **Banks** refers to Handbook of Simulation, by Jerry Banks, John Wiley & Sons, Inc., August 1998, ISBN 0-471-13403-1, page 3.

    SIMULATION: "Simulation is the imitation of the operation of a real-world process or system over time. Simulation involves the generation of an artificial history of the system and the observation of that artificial history to draw inferences concerning the operating characteristics of the real system that is represented".

Specification-objections-informalities

14. The Specification is objected to because of the following informalities. Appropriate correction is required.

15. CUMMULATIVE PROBABILITIES. Specification page 5 line 8 states "Each hierarchical instruction is assigned a probability such that a cumulative probability along any hierarchical path equals 1.0", which does not appear accurate. Specification page 6 line 19 states "the cumulative probability of the first segment is 1.00", which is slightly better, but still not clear.
16. In view of the example at specification pages 5-6, and in view of basic probability theory, the Examiner suggests the following more standard terminology: "Beginning at any node, the sum of probabilities of the immediate exiting branches (or paths) is 1.0. For example, beginning at the CPU node, the nonzero exiting branches are 0.25 + 0.25 + 0.5 = 1.0. Further, the probability of any specific machine instruction is the product of the probabilities of taking each branch that leads to said specific machine instruction. Note that the sum of the probabilities of all specific machine instructions should equal 1.0."
17. COMPLETE SET OF INSTRUCTIONS. Specification page 2 line 25 states "The random verification framework may be run continuously to test emulation of the complete set of instructions from the native ISA", and page 4 line 25 states "pseudo-random generation of machine level instructions to comprehensively test... random generation schemes", and page 6 line 23 "can insure that all binary instructions are eventually tested", and Abstract line 19 states "test emulation of the complete set of instructions from the native ISA".
18. From basic probability theory, it appears that the only way to be certain that all binary instructions are generated using pseudo-random generation requires two conditions: (1) all branching probabilities are greater than zero, and (2) the testing continues for an infinite length of time (or until such time as some bookkeeping mechanism determines that all binary instructions gave been generated at least once)..
19. BRANCHING. Specification page 6 lines 25-30 discusses using a seed to generate the probability sequence "0.34, 0.8, ...". Said 0.35 would generate a first level hierarchy branch of to CPU type instructions (from 0.0 to 1.0), then said 0.8 would generate a second level hierarchy branch to memory type instructions (from 0.50 to 1.0)." Note that each level of branching requires a random number (using the Applicant's procedure), and that the first level branching was omitted. Alternately, and probably more computationally efficient, after the probability of each individual machine instruction is calculated (as discussed above,

based on the product of the branching probabilities along that path), then a single random number can be used to directly determine the individual machine instruction. To summarize, Applicant's example at page 6 does not appear to be correct, because it omits the first branch, which branches to either CPU type instructions, or to FP type instructions according to the specification example.

20. GROUPING. Additionally, Applicant's grouping and sub-grouping of types of instructions is highly unorthodox, and possibly not logical. For example, at specification page 5 line 3, the first branching (or "segmentation" per Applicant's terminology) into "floating instructions and CPU instructions" is highly unusual because the FPU (floating point unit) is a subset of the CPU (central processing unit), according to Tucker at the top of page 413. See Tucker at the pages 413-414 for a more standard grouping and sub-grouping of instructions.
21. RANDOM. Specification page 7 line 10 states "generates pseudo-random native code machine language and generates a random initial machine state". It is not clear what difference is intended between the terms "pseudo-random" and "random". In general, different words should have different meanings, particularly in the same sentence.

35 USC § 101-statutory subject matter-Ex parte Lyell

22. 35 U.S.C. 101 reads as follows: Whoever invents or discovers any new and useful **process, machine, manufacture, or composition of matter,** or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.
23. **Claims 1-8 are rejected under 35 U.S.C. 101** because the claimed invention is directed to non-statutory subject matter.
24. Claim 1 preamble states "An apparatus for pseudo-random testing binary emulation, comprising:". Thus, claim 1 appears to intended as a "machine" (or apparatus) according to the statutory categories of 35 USC 101.
25. However, the claim 1 limitations include terms appear to be process steps: "generates... executes... emulates... compares". See MPEP 2173.05(p)(II), which states:

> A single claim which claims both an apparatus and the method steps of using the apparatus is indefinite under 35 U.S.C. 112, second paragraph. In Ex parte Lyell, 17 USPQ2d 1548 (Bd. Pat. App. & Inter. 1990), a claim directed to an automatic transmission workstand and the method steps of using it was held to be ambiguous and

properly rejected under 35 U.S.C. 112, second paragraph.

Such claims should also be rejected under 35 U.S.C. 101 based on the theory that the claim is directed to neither a "process" nor a "machine," but rather embraces or overlaps two different statutory classes of invention set forth in 35 U.S.C. 101 which is drafted so as to set forth the statutory classes of invention in the alternative only. Id. at 1551.

26. Thus, Claim 1 appears to be simultaneously directed at multiple 35 USC 101 statutory categories, and thus is rejected under 35 USC 101 and under 35 USC 112 as indefinite. Claims 2-8 are rejected for the same reason as base claim 1. For the purpose of examination, the Examiner will interpret said process steps as mere intended use.

35 USC § 112-Second Paragraph-indefinite claims

27. The following is a quotation of the second paragraph of 35 U.S.C. 112: The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

28. **Claims 1-8, and 19 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite** for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

29. The definitions listed above from Freedman and McGraw-Hill dictionaries are used throughout the claims.

30. In claim 1, some limitations include what appear to be process steps: **"that generates... that executes... that emulates... that compares".** Said process steps are interpreted as merely intended use.

31. In claim 1, the term **"binary instruction sequence"** is not clear . Specification page 6 lines 25 discusses using a seed to generate the "probability sequence", and said sequence is used to select a single instruction. Thus, it is not clear if the term "binary instruction sequence" refers to sequence of randomly generated numbers which is used to select a single instruction, or whether said term refers to a sequence of instructions.

32. Claims 2-8 are rejected for the same reason as base claim 1.

33. In claim 5, it is not clear how the term **"simulator"** is patentably distinct from the base claim 1 limitation "target architecture execution engine". These terms must be interpreted as patentably distinct in order for claim 5 for further limit its base claim 1. See 35 USC 103 rejection below further discussion Also see claim 19 immediately below.

34. In claim 19, **"the emulation of the native instruction sequence and the execution of the emulated native instruction sequence is completed on a binary instruction emulator operating on a simulator, which operates on the first platform"** is not clear. Note that the Examiner is interpreting the claim term "native" and the claim term "first" as equivalent to each other, and as equivalent to the Mitchell term "source", and equivalent to the Banks term "base". Thus, this claim appears to be performing an emulation of the native instruction sequence upon the native (first) platform. In other words, the native hardware appears to be emulating itself. Please clarify if this is the intended meaning of this claim. Further, please clarify the difference between "emulator" and "simulator" in this claim, because they appear to refer to the same element.

Claim Interpretation

35. **The claim language is interpreted in light of the specification.** Although the claims are interpreted in light of the specification, limitations from the specification are not read into the claims. See *In re Van Geuns,* 988 F.2d 1181, 26 USPQ2d 1057 (Fed. Cir. 1993).
36. In claim 1, the term **"binary instruction sequence"** is interpreted as "machine level instruction sequence".
37. In claim 1, the term **"native"** is interpreted as equivalent to the Mitchell term "source", and equivalent to the Banks term "base".
38. In claim 7 and throughout the claims, **"ISA"** is interpreted as "instruction set architecture", per Applicant's definition. Note that Applicant's definition for said acronym is different from the definitions in the Freedman Encyclopedia: ISA: "(1) (Industry Standard Architecture)... An expansion bus commonly used in PCs.... (2) (Interactive Services Association) A trade group for the online industry...". The Examiner suggests that the Applicant avoid the use of "ISA" in the claims, in order to avoid confusion.
39. In claim 16, the term **"first"** is interpreted as equivalent to the Mitchell term "source", and equivalent to the Banks term "base", and equivalent to the term "native" in claim 1.

Claim Rejections - 35 USC § 103

40. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action: (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject

matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

41. The factual inquiries set forth in *Graham* v. *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows: Determining the scope and contents of the prior art. Ascertaining the differences between the prior art and the claims at issue. Resolving the level of ordinary skill in the pertinent art. Considering objective evidence present in the application indicating obviousness or nonobviousness.

42. Claims 1-20 are rejected under 35 U.S.C. 103(a) as being unpatentable.

43. Claim 1 is rejected under 35 U.S.C. 103(a) as being unpatentable over Mitchell in view of Banks and Aharon.

44. Independent claim 1 is an "apparatus" (or "machine") claim with 4 limitations, numbered by the Examiner for clarity.

45. [3] **"a target architecture execution engine that executes the binary instruction sequence to produce a final state S2, the target architecture execution engine comprising a binary emulator that emulates the binary instruction sequence according to the target architecture"** is disclosed by Mitchell column 1 line 32 "Emulation is the imitation of the operation of a first ("source") CPU by a second ("target") CPU. The target CPU is specially programmed and architected to permit it to execute programs written for the source CPU. A program written for the source CPU comprises a sequence of source instructions which are provided, one-by-one to the target CPU. The target CPU responds to each source instruction by executing one or more target instructions".

46. The remaining limitations are not disclosed by Mitchell.

47. [2] **"a native architecture engine that executes the binary instruction sequence to produce a final state S1"** is disclosed by Banks page 17 "Validation is the determination that the conceptual model is an accurate representation of the real system. Can the model be substituted for the real system for the purposes of experimentation? If there is an existing system, call it the base system, and ideal way to validate the model is to compare its output to that of the base system." Note that Banks' "real... existing system" discloses "native architecture engine".

48. [4] **"a verification engine that compares the final state S1 and the final state S2, wherein the final state S1 and the final state S2 do not match, an emulation failure has occurred"** is disclosed by Banks page 17 "Validation is the determination that the conceptual model is an accurate representation of the real system. Can the model be substituted for the real system for the purposes of experimentation? If there is an existing system, call it the base system, and ideal way to validate the model is to compare its output to that of the base system."
49. [1] **"a random code generator that generates an initial machine state and a binary instruction sequence"** is disclosed by Aharon column 1 line 21 "dynamic biased pseudo-random test pattern generator (RTPG) for the functional verification of hardware designs".
50. **At the time** the invention was made, it would have been obvious to a person of ordinary skill in the art to use Banks and Aharon to modify Mitchell. One of ordinary skill in the art, starting with Mitchell's emulator, would be motivated to make certain that the emulator was working properly by validating it using Banks' "ideal" method of comparing its output against the base system, if a base system was available. Further, one of ordinary skill in the art would be motivated to save time and money during the validation process by using the same procedure that was used to test functional behavior in the "base" (or "source" or "native") system, specifically Aharon's RTPG to generate inputs (initial machine state and a binary instruction sequence).
51. Claims 2-8 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mitchell in view of Banks and Aharon.
52. Claims 2-8 depend from claim 1, with the following additional limitations.
53. In claim 2, **"the native and the target architecture execution engines communicate using machine-to-machine communications protocols"** is disclosed by Banks page 17 "ideal way to validate the model is to compare its output to that of the base system." One of ordinary skill in the art would interpret Bank's term "compare" broadly as including comparing using the almost universal standard communications protocol TCP/IP. Note that Freedman Encyclopedia defines TCP/IP as "(Transmission Control Protocol/Internet Protocol) A communications protocol... to internetwork dissimilar systems. It is a de facto UNIX standard, but is supported on almost all platforms. TCP/IP is the protocol of the Internet".

54. Note that reasonable "inferences", and "common sense" may be considered in formulating rejections for obviousness. Specifically, *In re Preda*, 401 F.2d 825, 159 USPQ 342, 344 (CCPA 1968) states "in considering the disclosure of a reference, it is proper to take into account not only specific teachings of the reference but also the inferences which one skilled in the art would reasonably be expected to draw therefrom." Also, *In re Bozek*, 416 F.2d 738, 163 USPQ 545, 549 (CCPA 1969) states that obviousness may be concluded from "common knowledge and common sense of the person of ordinary skill in the art without any specific hint or suggestion in a particular reference". Additionally, see *In re Gauerke*, 24 CCPA 725, 86 F.2d 330, 31 USPQ 330, 333 (CCPA 1936), and *In re Libby*, 45 CCPA 944, 255 F.2d 412, 118 USPQ 94, 96 (CCPA 1958), and *In re Jacoby*, 309 F.2d 738, 125 USPQ 317, 319 (CCPA 1962), and *In re Wiggins*, 488 F.2d 538, 543, 1979 USPQ 421, 424 (CCPA 1973).
55. In claim 3, **"the communications protocol is a TCP/IP protocol"** is disclosed by Banks page 17 "ideal way to validate the model is to compare its output to that of the base system." One of ordinary skill in the art would interpret Bank's term "compare" broadly as including comparing using the almost universal standard communications protocol TCP/IP. Note that Freedman Encyclopedia defines TCP/IP as "(Transmission Control Protocol/Internet Protocol) A communications protocol... to internetwork dissimilar systems. It is a de facto UNIX standard, but is supported on almost all platforms. TCP/IP is the protocol of the Internet".
56. In claim 4, **"when the emulation failure occurs, the information related to the emulation failure is written to a file"** is disclosed by Banks page 17 "ideal way to validate the model is to compare its output to that of the base system." One of ordinary skill in the art would interpret Bank's term "compare" broadly as including an "error handling" routine that, at bare minimum, created a list or file of failures. Note that Freedman Encyclopedia defines "error handling" as "Routines in a program that respond to errors. The measurement of quality in error handling is based on how the system informs the user of such conditions and what alternatives it provides for dealing with them".
57. In claim 5, **"The target platform is a simulator"** is disclosed by Mitchell column 1 line 32 "Emulation is the imitation of the operation of a first ("source") CPU by a second ("target")

CPU. The target CPU is specially programmed and architected to permit it to execute programs written for the source CPU. A program written for the source CPU comprises a sequence of source instructions which are provided, one-by-one to the target CPU. The target CPU responds to each source instruction by executing one or more target instructions". Note that Freedman Encyclopedia defines "simulation" as "(1) The mathematical representation of the interaction of real-world objects. *See scientific applications.* (2) The execution of a machine language program designed to run in a foreign computer." Italics in original. Also, the McGraw-Hill Dictionary defines "simulate" as "[ENG] To mimic some or all of the behavior of one system with a different dissimilar system, particularly with computers, models, or equipment."

58. It is possible that the Applicant is intending to claim software through this limitation. Note that "software" is generally not patentable as such, but must be claimed as process steps, or as machine, or as manufacture. For example, software may be claimed as machine or [article of] manufacture by stating "a computer readable media containing instructions, which when executed, cause the following steps to be performed". Also see MPEP 2106 regarding computer related inventions.
59. In claim 6, **"the target platform is a hardware embodiment"** is disclosed by Mitchell column 1 line 32 "Emulation is the imitation of the operation of a first ("source") CPU by a second ("target") CPU. The target CPU is specially programmed and architected to permit it to execute programs written for the source CPU. A program written for the source CPU comprises a sequence of source instructions which are provided, one-by-one to the target CPU. The target CPU responds to each source instruction by executing one or more target instructions".
60. In claim 7, **"the random code generator comprises a probability file comprising probability values for each instruction in a native instructions set architecture (ISA)"** is disclosed by Aharon FIG 2 "BIASING STRATEGIES" and "RTPG" and column 6 line 19 "biasing strategy".
61. In claim 8, **"the probability values in the probability file are user-generated"** is disclosed by Aharon FIG 2 "BIASING STRATEGIES" and "RTPG" and column 6 line 19 "biasing strategy".

62. MOTIVATION FOR DEPENDENT CLAIMS 2-8. **At the time** the invention was made, it would have been obvious to a person of ordinary skill in the art to use Banks and Aharon to modify Mitchell. One of ordinary skill in the art, starting with Mitchell's emulator, would be motivated to make certain that the emulator was working properly by validating it using Banks' "ideal" method of comparing its output against the base system, if a base system was available. Further, one of ordinary skill in the art would be motivated to save time and money during the validation process by using the same procedure that was used to test functional behavior in the "base" (or "source" or "native") system, specifically Aharon's RTPG to generate inputs (initial machine state and a binary instruction sequence). Additionally, one of ordinary skill in the art would be motivated to efficiently implement Bank's "compare" by using standard communication protocols that are supported on almost all platforms, and to bias the RTPG in various ways in order to evaluate the instructions of immediate interest.
63. Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over Mitchell in view of Banks and Aharon.
64. Independent claim 9 is a "method" (or "process") claim with 8 limitations, numbered by the Examiner for clarity.
65. [3] **"initializing a native machine according to the initial machine state"** is disclosed by Mitchell column 1 line 32 "Emulation is the imitation of the operation of a first ("source") CPU by a second ("target") CPU. The target CPU is specially programmed and architected to permit it to execute programs written for the source CPU. A program written for the source CPU comprises a sequence of source instructions which are provided, one-by-one to the target CPU. The target CPU responds to each source instruction by executing one or more target instructions". One of ordinary skill in the art would broadly interpret "sequence of source instructions" as including initialization instructions, because CPUs are sequential circuits (the output depends upon the input and the initial state). In other words, the CPUs have memory, and the initial state of the memory affects the output for any given input.
66. [5] **"initializing a target machine according to the initial machine state"** is disclosed by Mitchell column 1 line 32 "Emulation is the imitation of the operation of a first ("source") CPU by a second ("target") CPU. The target CPU is specially programmed and architected

to permit it to execute programs written for the source CPU. A program written for the source CPU comprises a sequence of source instructions which are provided, one-by-one to the target CPU. The target CPU responds to each source instruction by executing one or more target instructions". One of ordinary skill in the art would broadly interpret "sequence of source instructions" as including initialization instructions, because CPUs are sequential circuits (the output depends upon the input and the initial state). In other words, the CPUs have memory, and the initial state of the memory affects the output for any given input.

67. Mitchell does not expressly disclose the additional limitations.

68. [1] **"generating a pseudo-random binary instruction sequence"** is disclosed by Aharon column 1 line 21 "dynamic biased pseudo-random test pattern generator (RTPG) for the functional verification of hardware designs".

69. [2] **"generating an initial machine state"** is disclosed by Aharon column 1 line 21 "dynamic biased pseudo-random test pattern generator (RTPG) for the functional verification of hardware designs". Note that for sequential hardware circuits such as CPUs, the initial state of the sequential circuit must be defined as part of the functional verification, because the output is dependent upon the input and the initial state of the sequential circuit. In other words, the CPUs have memory, and the initial state of the memory affects the output for any given input.

70. [4] **"executing the binary instruction sequence on the native machine to produce a final state S1"** is disclosed by Banks page 17 "Validation is the determination that the conceptual model is an accurate representation of the real system. Can the model be substituted for the real system for the purposes of experimentation? If there is an existing system, call it the base system, and ideal way to validate the model is to compare its output to that of the base system."

71. [6] **"emulating the binary instruction sequence on the target machine"** is disclosed by Banks page 17 "Validation is the determination that the conceptual model is an accurate representation of the real system. Can the model be substituted for the real system for the purposes of experimentation? If there is an existing system, call it the base system, and ideal way to validate the model is to compare its output to that of the base system."

72. [7] **"executing the emulated binary instruction sequence to produce a final state S2"** is disclosed by Banks page 17 "Validation is the determination that the conceptual model is an accurate representation of the real system. Can the model be substituted for the real system for the purposes of experimentation? If there is an existing system, call it the base system, and ideal way to validate the model is to compare its output to that of the base system."
73. [8] **"comparing the final state S1 to the final state S2 to determine an emulation error"** disclosed by Banks page 17 "Validation is the determination that the conceptual model is an accurate representation of the real system. Can the model be substituted for the real system for the purposes of experimentation? If there is an existing system, call it the base system, and ideal way to validate the model is to compare its output to that of the base system."
74. **At the time** the invention was made, it would have been obvious to a person of ordinary skill in the art to use Banks and Aharon to modify Mitchell. One of ordinary skill in the art, starting with Mitchell's emulator, would be motivated to make certain that the emulator was working properly by validating it using Banks' "ideal" method of comparing its output against the base system, if a base system was available. Further, one of ordinary skill in the art would be motivated to save time and money during the validation process by using the same procedure that was used to test functional behavior in the "base" (or "source" or "native") system, specifically Aharon's RTPG to generate inputs (initial machine state and a binary instruction sequence).
75. Claims 10-15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mitchell in view of Banks and Aharon.
76. Claims 10-15 depend from claim 9, with the following additional limitations.
77. In claim 10, **"communicating the initial machine state and the binary instruction sequence to the target machine using a machine-to-machine communications protocol"** is disclosed by Banks page 17 "ideal way to validate the model is to compare its output to that of the base system." One of ordinary skill in the art would interpret Bank's term "compare" broadly as including comparing using the almost universal standard communications protocol TCP/IP. Note that Freedman Encyclopedia defines TCP/IP as "(Transmission Control Protocol/Internet Protocol) A communications protocol... to

internetwork dissimilar systems. It is a de facto UNIX standard, but is supported on almost all platforms. TCP/IP is the protocol of the Internet".

78. Also in claim 10, **"communicating the final state S2 to the native machine using the machine-to-machine communication protocol"** is disclosed by Banks page 17 "ideal way to validate the model is to compare its output to that of the base system." One of ordinary skill in the art would interpret Bank's term "compare" broadly as including comparing using the almost universal standard communications protocol TCP/IP. Note that Freedman Encyclopedia defines TCP/IP as "(Transmission Control Protocol/Internet Protocol) A communications protocol... to internetwork dissimilar systems. It is a de facto UNIX standard, but is supported on almost all platforms. TCP/IP is the protocol of the Internet".
79. In claim 11, **"the machine-to-machine protocol is a TCP/IP protocol"** is disclosed by Banks page 17 "ideal way to validate the model is to compare its output to that of the base system." One of ordinary skill in the art would interpret Bank's term "compare" broadly as including comparing using the almost universal standard communications protocol TCP/IP. Note that Freedman Encyclopedia defines TCP/IP as "(Transmission Control Protocol/Internet Protocol) A communications protocol... to internetwork dissimilar systems. It is a de facto UNIX standard, but is supported on almost all platforms. TCP/IP is the protocol of the Internet".
80. In claim 12, **"storing the information related to the emulation failure, the information comprising the initial machine state, the binary instruction sequence and the final states S1 and S2"** is disclosed by Banks page 17 "ideal way to validate the model is to compare its output to that of the base system." One of ordinary skill in the art would interpret Bank's term "compare" broadly as including an "error handling" routine that, at bare minimum, created a list or file of failures. Note that Freedman Encyclopedia defines "error handling" as "Routines in a program that respond to errors. The measurement of quality in error handling is based on how the system informs the user of such conditions and what alternatives it provides for dealing with them". Note that the initial machine states, and the instructions, and the final states are the minimum information required for comparing the functionality of sequential circuits such as CPUs, and are the minimum information required for analyzing

failures (errors) when the model (target) output is different from the base (native) system output.

81. In claim 13, **"the binary emulation is executed on a simulator"** is disclosed by Mitchell column 1 line 32 "Emulation is the imitation of the operation of a first ("source") CPU by a second ("target") CPU. The target CPU is specially programmed and architected to permit it to execute programs written for the source CPU. A program written for the source CPU comprises a sequence of source instructions which are provided, one-by-one to the target CPU. The target CPU responds to each source instruction by executing one or more target instructions". Note that Freedman Encyclopedia defines "simulation" as "(1) The mathematical representation of the interaction of real-world objects. *See scientific applications*. (2) The execution of a machine language program designed to run in a foreign computer." Italics in original. Also, the McGraw-Hill Dictionary defines "simulate" as "[ENG] To mimic some or all of the behavior of one system with a different dissimilar system, particularly with computers, models, or equipment."
82. It is possible that the Applicant is intending to claim software through this limitation. Note that "software" is generally not patentable as such, but must be claimed as process steps, or as machine, or as manufacture. For example, software may be claimed as machine or [article of] manufacture by stating "a computer readable media containing instructions, which when executed, cause the following steps to be performed". Also see MPEP 2106 regarding computer related inventions.
83. In claim 14, **"the binary emulation is executed on a hardware device"** is disclosed by Mitchell column 1 line 32 "Emulation is the imitation of the operation of a first ("source") CPU by a second ("target") CPU. The target CPU is specially programmed and architected to permit it to execute programs written for the source CPU. A program written for the source CPU comprises a sequence of source instructions which are provided, one-by-one to the target CPU. The target CPU responds to each source instruction by executing one or more target instructions".
84. In claim 15, **"pseudo-randomly generating a probability value"** is disclosed by Aharon FIG 2 "BIASING STRATEGIES" and "RTPG" and column 6 line 19 "biasing strategy".

85. Also in claim 15, **"selecting the binary instruction sequence based on the probability value"** is disclosed by Aharon FIG 2 "BIASING STRATEGIES" and "RTPG" and column 6 line 19 "biasing strategy".
86. MOTIVATION FOR DEPENDENT CLAIMS 10-15. **At the time** the invention was made, it would have been obvious to a person of ordinary skill in the art to use Banks and Aharon to modify Mitchell. One of ordinary skill in the art, starting with Mitchell's emulator, would be motivated to make certain that the emulator was working properly by validating it using Banks' "ideal" method of comparing its output against the base system, if a base system was available. Further, one of ordinary skill in the art would be motivated to save time and money during the validation process by using the same procedure that was used to test functional behavior in the "base" (or "source" or "native") system, specifically Aharon's RTPG to generate inputs (initial machine state and a binary instruction sequence). Additionally, one of ordinary skill in the art would be motivated to efficiently implement Bank's "compare" by using standard communication protocols that are supported on almost all platforms, and to bias the RTPG in various ways in order to evaluate the instructions of immediate interest.
87. Claim 16 is rejected under 35 U.S.C. 103(a) as being unpatentable over Mitchell in view of Banks and Aharon.
88. Independent claim 16 is a "method" (or "process") claim with 7 limitations, numbered by the Examiner for clarity.
89. [2] **"providing the native instruction sequence and the initial machine state to a first platform having a first instruction set architecture (ISA)"** is disclosed by Mitchell column 1 line 32 "Emulation is the imitation of the operation of a first ("source") CPU by a second ("target") CPU. The target CPU is specially programmed and architected to permit it to execute programs written for the source CPU. A program written for the source CPU comprises a sequence of source instructions which are provided, one-by-one to the target CPU. The target CPU responds to each source instruction by executing one or more target instructions". One of ordinary skill in the art would broadly interpret "sequence of source instructions" as including initialization instructions, because CPUs are sequential circuits (the

output depends upon the input and the initial state). In other words, the CPUs have memory, and the initial state of the memory affects the output for any given input.

90. [3] **"providing the native instruction sequence to a second platform having a second ISA"** is disclosed by Mitchell column 1 line 32 "Emulation is the imitation of the operation of a first ("source") CPU by a second ("target") CPU. The target CPU is specially programmed and architected to permit it to execute programs written for the source CPU. A program written for the source CPU comprises a sequence of source instructions which are provided, one-by-one to the target CPU. The target CPU responds to each source instruction by executing one or more target instructions". One of ordinary skill in the art would broadly interpret "sequence of source instructions" as including initialization instructions, because CPUs are sequential circuits (the output depends upon the input and the initial state). In other words, the CPUs have memory, and the initial state of the memory affects the output for any given input.
91. [4] **"emulating the native instruction sequence according to the second ISA"** is disclosed by Mitchell column 1 line 32 "Emulation is the imitation of the operation of a first ("source") CPU by a second ("target") CPU. The target CPU is specially programmed and architected to permit it to execute programs written for the source CPU. A program written for the source CPU comprises a sequence of source instructions which are provided, one-by-one to the target CPU. The target CPU responds to each source instruction by executing one or more target instructions". One of ordinary skill in the art would broadly interpret "sequence of source instructions" as including initialization instructions, because CPUs are sequential circuits (the output depends upon the input and the initial state). In other words, the CPUs have memory, and the initial state of the memory affects the output for any given input.
92. Mitchell does not disclose the additional limitations.
93. [1] **"randomly generating a native instruction sequence and an initial machine state"** is disclosed by Aharon column 1 line 21 "dynamic biased pseudo-random test pattern generator (RTPG) for the functional verification of hardware designs". One of ordinary skill in the art would broadly interpret "test pattern generator" as including initialization instructions creating an initial machine state for sequential hardware. Note that CPUs are sequential

circuits (the output depends upon the input and the initial state). In other words, the CPUs have memory, and the initial state of the memory affects the output for any given input.

94. [5] **"executing the native instruction sequence in the first platform, the execution providing a final state S1"** is disclosed by Banks page 17 "Validation is the determination that the conceptual model is an accurate representation of the real system. Can the model be substituted for the real system for the purposes of experimentation? If there is an existing system, call it the base system, and ideal way to validate the model is to compare its output to that of the base system."

95. [6] **"executing the emulated native instruction sequence on the second platform, the execution providing a final state S2"** is disclosed by Banks page 17 "Validation is the determination that the conceptual model is an accurate representation of the real system. Can the model be substituted for the real system for the purposes of experimentation? If there is an existing system, call it the base system, and ideal way to validate the model is to compare its output to that of the base system."

96. [7] **"comparing the final states S1 and S2 to determine an emulation failure"** is disclosed by Banks page 17 "Validation is the determination that the conceptual model is an accurate representation of the real system. Can the model be substituted for the real system for the purposes of experimentation? If there is an existing system, call it the base system, and ideal way to validate the model is to compare its output to that of the base system."

97. **At the time** the invention was made, it would have been obvious to a person of ordinary skill in the art to use Banks and Aharon to modify Mitchell. One of ordinary skill in the art, starting with Mitchell's emulator, would be motivated to make certain that the emulator was working properly by validating it using Banks' "ideal" method of comparing its output against the base system, if a base system was available. Further, one of ordinary skill in the art would be motivated to save time and money during the validation process by using the same procedure that was used to test functional behavior in the "base" (or "source" or "native") system, specifically Aharon's RTPG to generate inputs (initial machine state and a binary instruction sequence).

98. Claims 17-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Mitchell in view of Banks and Aharon.

99. Claims 17-20 depend from claim 16, with the following additional limitations.

100. In claim 17, **"the native instruction sequence is randomly generated from a set of all instructions in the first ISA"** is disclosed by Aharon column 1 line 21 "dynamic biased pseudo-random test pattern generator (RTPG) for the functional verification of hardware designs", and FIG 2 "BIASING STRATEGIES" and "RTPG", and column 6 line 19 "biasing strategy".

101. In claim 18, **"the random generation is based on a user-defined value assigned to each instruction in the first ISA"** is disclosed by Aharon column 1 line 21 "dynamic biased pseudo-random test pattern generator (RTPG) for the functional verification of hardware designs", and FIG 2 "BIASING STRATEGIES" and "RTPG", and column 6 line 19 "biasing strategy".

102. In claim 19, **"the emulation of the native instruction sequence and the execution of the emulated native instruction sequence is completed on a binary instruction emulator operating on a simulator, which operates on the first platform"** is disclosed by Mitchell column 1 line 32 "Emulation is the imitation of the operation of a first ("source") CPU by a second ("target") CPU. The target CPU is specially programmed and architected to permit it to execute programs written for the source CPU. A program written for the source CPU comprises a sequence of source instructions which are provided, one-by-one to the target CPU. The target CPU responds to each source instruction by executing one or more target instructions".

103. In claim 20, **"the emulation is completed on a binary instruction emulator operating on the second platform"** is disclosed by Mitchell column 1 line 32 "Emulation is the imitation of the operation of a first ("source") CPU by a second ("target") CPU. The target CPU is specially programmed and architected to permit it to execute programs written for the source CPU. A program written for the source CPU comprises a sequence of source instructions which are provided, one-by-one to the target CPU. The target CPU responds to each source instruction by executing one or more target instructions".

104. MOTIVATION FOR DEPENDENT CLAIMS 17-20. **At the time** the invention was made, it would have been obvious to a person of ordinary skill in the art to use Banks and Aharon to modify Mitchell. One of ordinary skill in the art, starting with Mitchell's

emulator, would be motivated to make certain that the emulator was working properly by validating it using Banks' "ideal" method of comparing its output against the base system, if a base system was available. Further, one of ordinary skill in the art would be motivated to save time and money during the validation process by using the same procedure that was used to test functional behavior in the "base" (or "source" or "native") system, specifically Aharon's RTPG to generate inputs (initial machine state and a binary instruction sequence) according to Aharon's biasing strategy. Additionally, one of ordinary skill in the art would be motivated to efficiently implement Bank's "compare" by using standard communication protocols that are supported on almost all platforms, and to bias the RTPG in various ways in order to evaluate the instructions of immediate interest.

Additional Cited Prior Art

The following US patents or publications are hereby cited as prior art, but have not been used for rejection. Applicant should review these carefully before responding to this office action.

US Patent 6,009,261 by Scalzi Abstract discloses: "program translation and execution... incompatible architecture".

US Patent 6,606,721 by Gowin column 2 line 53 et. seq. discloses:

> For example, generating stimuli based solely upon current machine state information may not adequately test some complex processors such as Very Long Instruction Word ("VLIW") and Reduced Instruction Set Computer ("RISC") processors, which may have special rules regarding usage of certain resources. Similarly, some processor architectures allow certain groups of instructions to execute at the same architectural time. Instructions that are dynamically generated and simulated in a pseudo-random test pattern must therefore adhere to the processor's instruction grouping rules. This creates another case where the creation of input stimulus is not just a function of the current state, and therefore requires additional components in the test generator to restrict the grouping of instructions and possible operands to a given set of rules. In addition, while a dynamically generated test setup may run flawlessly on a single processor model, the same test may violate memory coherence rules when run on a multiprocessor model. Finally, the creation of highly deterministic code sequences, such as sequences that execute if then else statements and other conditional control breaks, can be problematic in dynamically generated tests. The present invention addresses the limitations of current-technology dynamic psuedo-random test pattern generators, which currently select and generate instructions based only upon current machine state. The present invention is a method and apparatus that is useful in a pseudo-random test generation setup because the present invention dynamically tracks selected system resources in a golden model, thus enabling the selection, scheduling, and simulation of

instructions when generating a test to be based upon current and future system resource availability, current architectural state of the golden model, and the resource utilization rules appropriate for the architecture of the system under test. When the test generator selects an instruction for simulation on the golden model, the present invention assesses the system resources that the execution of the instruction will require as a function of time (processor cycles) and marks those resources as in use or unavailable for the appropriate amount of time. Tracking selected system resources in this manner increases the efficiency of a dynamic test generator, because the selection of instructions that require resources that are unavailable or that conflict with other pending instructions can be avoided until the resource or conflict clears. Moreover, the capability to track system resources enables the developers of dynamic pseudo-random test generators to incorporate and utilize special resource-utilization rules that some processor architectures may have, thus enabling the generation of more robust tests for these architectures. Dynamic pseudo-random test generators that include the present invention thus provide a better functional verification test of some processors, because such test generators are capable of taking into account machine state, resource availability, and instruction grouping rules in selecting and generating pseudo-random test stimuli.

Conclusion

All claims stand rejected.

Communication

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Eduardo Garcia-Otero whose telephone number is 703-305-0857. The examiner can normally be reached on Tuesday through Friday from 9:00 AM to 8:00 PM. If attempts to reach the Examiner by telephone are unsuccessful, the Examiner's supervisor, Kevin Teska, can be reached at (703) 305-9704. The fax phone number for this group is 703-872-9306. Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the group receptionist, whose telephone number is (703) 305-3900.

* * * *

KEVIN J. TESKA
SUPERVISORY
PATENT EXAMINER